(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 December 2001 (13.12.2001)

PCT

(10) International Publication Number
WO 01/095631 A3

(51) International Patent Classification[7]: H04N 5/278

(21) International Application Number: PCT/GB01/02547

(22) International Filing Date: 11 June 2001 (11.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0014161.4 9 June 2000 (09.06.2000) GB
0024413.7 5 October 2000 (05.10.2000) GB

(71) Applicant *(for all designated States except US)*: **BRITISH BROADCASTING CORPORATION** [GB/GB]; Broadcasting House, London W1A 1AA (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KIRBY, David, Graham** [GB/GB]; 21 Wallace Fields, Epsom, Surrey KT17 3AX (GB). **POOLE, Christopher, Edward** [GB/GB]; Flat 5, Oak House, Oakfield Drive, Reigate, Surrey RH2 9NY (GB). **WIEWIORKA, Adam** [GB/GB]; 16 Ashbourne Grove, Chiswick, London W4 2JH (GB). **LAHR, William, Oscar** [GB/GB]; 19 Cromford Road, London SW18 1NZ (GB).

(74) Agent: **ROBSON, Aidan, John**; Reddie & Grose, 16 Theobalds Road, London WC1X 8PL (GB).

(81) Designated States *(national)*: CA, GB, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *with international search report*

(88) Date of publication of the international search report:
6 September 2002

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/095631 A3

(54) Title: GENERATION SUBTITLES OR CAPTIONS FOR MOVING PICTURES

(57) Abstract: A method for generating subtitles for audiovisual material received and analyses a text file containing dialogue spoken in audiovisual material and provides a signal representative of the text. The text information and audio signal are aligned in time using time alignment speech recognition and the text and timing information are then output to a subtitle file. Colours can be assigned to different speakers or groups of speakers. Subtitles are derived by receiving and analysing a text file containing dialogue spoken by considering each word in turn and the next information signal, assigning a score to each subtitle in a plurality of different possible subtitle formatting options which lead to that word. The steps are then repeated until all the words in the text information signal have been used and the subtitle formatting option which gives the best overall score is then derived.

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	"COMPUTER-AIDED DIALOG SYNCHRONIZATION AND SUBTITLE GENERATION FOR VIDEO SYSTEMS" IBM TECHNICAL DISCLOSURE BULLETIN, IBM CORP. NEW YORK, US, vol. 36, no. 8, 1 August 1993 (1993-08-01), pages 119-120, XP000390163 ISSN: 0018-8689 the whole document	1,16-18, 20
A	EP 0 899 719 A (DIGITAL EQUIPMENT CORP) 3 March 1999 (1999-03-03) column 1, line 11 - line 52 column 3, line 6 -column 7, line 24	1-3, 16-18
A	STARK H ET AL: "BAYES THEOREM AND APPLICATIONS" , PROBABILITY, RANDOM PROCESSES AND ESTIMATION THEORY FOR ENGINEERS, PRENTICE-HALL, UPPER SADDLE RIVER, NJ, US, VOL. EDI-2, PAGE(S) 19-21 XP002922983 page 19, line 20 - line 22	2
A	KING C M ET AL: "DIGITAL CAPTIONING: EFFECTS OF COLOR-CODING AND PLACEMENT IN SYNCHRONIZED TEXT-AUDIO PRESENTATIONS" PROCEEDINGS OF ED-MEDIA. WORLD CONFERENCE ON EDUCATIONAL MULTIMEDIA NAD HYPERMEDIA, XX, XX, 25 June 1994 (1994-06-25), pages 329-334, XP001034702 page 331, line 29 - line 40 page 333, line 23 - line 37	4-8
A	GB 2 114 407 A (NAT RES DEV) 17 August 1983 (1983-08-17) page 1, line 46 -page 2, line 19 page 3, line 25 -page 13, line 7	29-31
A	FOX H ET AL: "Learning to extract and classify names from text" SYSTEMS, MAN, AND CYBERNETICS, 1998. 1998 IEEE INTERNATIONAL CONFERENCE ON SAN DIEGO, CA, USA 11-14 OCT. 1998, NEW YORK, NY, USA,IEEE, US, 11 October 1998 (1998-10-11), pages 1668-1673, XP010311001 ISBN: 0-7803-4778-1 page 1668, left-hand column, line 3 -page 1671, left-hand column, line 4	11

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-3, 16-18 (depending on claims 1-3), 20

   Method of generating subtitles providing text and associated timing information in an output file

2. Claims: 4-8, 16-18 (depending on claims 4-8)

   Method of assigning colour representative of different speacker to subtitles

3. Claims: 9,10, 16-18 (depending on claims 19 or 10)

   Method of detecting scene changes in audiovisual material

4. Claims: 11-14, 16-18 (depending on claims 11-14)

   Method of parsing electronic text file to identify different components thereof

5. Claims: 15, 16-18 (depending on claim 15), 21, 22

   Method of placing subtitles related to spech from speakers in a (moving) picture

6. Claim : 19

   Method of generating subtitles for audiovisual material without using a typed text file

7. Claims: 23- 28

   Method for generating subtitles for audiovisual material using a change of speaker indicating signal

8. Claims: 29-31

   method for automatic and efficent formatting of subtitles

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 December 2001 (13.12.2001)

PCT

(10) International Publication Number
WO 01/95631 A2

(51) International Patent Classification[7]: H04N 7/26

(21) International Application Number: PCT/GB01/02547

(22) International Filing Date: 11 June 2001 (11.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0014161.4 9 June 2000 (09.06.2000) GB
0024413.7 5 October 2000 (05.10.2000) GB

(71) Applicant *(for all designated States except US)*: **BRITISH BROADCASTING CORPORATION** [GB/GB]; Broadcasting House, London W1A 1AA (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KIRBY, David, Graham** [GB/GB]; 21 Wallace Fields, Epsom, Surrey KT17 3AX (GB). **POOLE, Christopher, Edward** [GB/GB]; Flat 5, Oak House, Oakfield Drive, Reigate, Surrey RH2 9NY (GB). **WIEWIORKA, Adam** [GB/GB]; 16 Ashbourne Grove, Chiswick, London W4 2JH (GB). **LAHR, William, Oscar** [GB/GB]; 19 Cromford Road, London SW18 1NZ (GB).

(74) Agent: **ROBSON, Aidan, John**; Reddie & Grose, 16 Theobalds Road, London WC1X 8PL (GB).

(81) Designated States *(national)*: CA, GB, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/95631 A2

(54) Title: GENERATION SUBTITLES OR CAPTIONS FOR MOVING PICTURES

(57) Abstract: A method for generating subtitles for audiovisual material received and analyses a text file containing dialogue spoken in audiovisual material and provides a signal representative of the text. The text information and audio signal are aligned in time using time alignment speech recognition and the text and timing information are then output to a subtitle file. Colours can be assigned to different speakers or groups of speakers. Subtitles are derived by receiving and analysing a text file containing dialogue spoken by considering each word in turn and the next information signal, assigning a score to each subtitle in a plurality of different possible subtitle formatting options which lead to that word. The steps are then repeated until all the words in the text information signal have been used and the subtitle formatting option which gives the best overall score is then derived.

Much of the time taken in preparing subtitles is spent in synchronising the text to the dialogue. If a subtitle appears or ends at a significantly different time from its associated dialogue, then this is distracting for viewers, and even more so for those with hearing impairments who may also be lip-reading. Hence, as the subtitles are being created, significant time is taken in ensuring that this aspect is correct.

As can be seen, current techniques to prepare subtitles are very labour-intensive and time-consuming. It is typical for it to take between 12 and 16 hours for each hour of programme being subtitled.

It has been proposed to use speech recognition to produce the text of what was spoken in an automatic or semi-automatic fashion. However, we have found that this does not work in practice, with even the best currently-available speech recognition techniques. The recordings are not made with speech recognition in mind, and the manner and variety of speech as well as the background noise are such that at times the speech recognition is so poor that the subtitle text is nonsense. Speech recognition has therefore been dismissed as being inappropriate at the present time.

Speech recognition is known for use in a variety of different ways. These include generating text from an audio file (United Kingdom Patent Specification GB 2 289 395A), editing video (Japanese Patent Application 09-091928 of 1997), controlling video (Japanese Patent Application 09-009199 of 1997), indexing video material (Wactlar et al., "Intelligent Access to Digital Video : Infomedia Project" Computer, May 1996, pages 46 to 52; Brown et al., "Open-Vocabulary Speech Indexing for Voice

speaker always appears in that same colour. Other speakers can be assigned that same colour (although this may sometimes not be permitted); however, apart from the colour white, text of the same colour but from different speakers must not appear in the same subtitle. Assigning colours in this way is a much more complex task for the subtitler as they must ensure that speakers do not appear together at any point in the programme before assigning them the same colour. If this is not done then there is the possibility that, should the two speakers subsequently appear together in the same subtitle, all the colours will need to be assigned in a different way and the subtitles completed so far changed to adopt the new colours.

A second aspect of this invention is directed to this problem of efficiently allocating colours to speakers, in a manner such that it can be undertaken automatically.

In implementing subtitling systems along the lines described below, it is desirable to be able to detect scene changes. This is of great assistance in colour allocation in particular. Scene change detection (as opposed to shot change detection) requires complex analysis of the video content and is difficult to achieve.

In accordance with a third aspect of this invention we provide a method of scene change detection which is relatively simple to implement but which nevertheless provides effective scene change detection for the purposes required.

The saving in time is such that where no script is available, it may be quicker to type a script so that the invention can be used, rather than proceeding with the conventional method. The script need not be printed but can just be typed as text into a computer file.

In the second aspect of the invention, a system for allocating colours to speakers is proposed, in which groups of speakers are formed, where each 'group' (which may only comprise one speaker) contains speakers who can be represented by the same colour. This typically produces a large plurality of groups. The available colours are then assigned to the groups such that all the speakers are allocated a colour. This is essentially done by ordering the groups, subject to appropriate overall criteria, before performing a search for an optimum colour scheme.

In the third aspect of the invention scene changes in audiovisual material are detected by identifying when speakers are active in the material, and detecting points in time in the dialogue where the group of active speakers changes.

An area of difficulty in working with printed scripts for use in the first aspect of the method is that many different script formats are in use, and it is difficult to automatically interpret the scripts so as to distinguish speech from speaker's name, titles, instructions to the director and actors, timing information, and so on.

In accordance with a fourth aspect of this invention we propose analysing or parsing a script by the use of a statistical method based on Bayes' theorem.

The text produced by the speech recogniser is already aligned to the spoken text, i.e. the microphone output, and the video of the audiovisual material is already aligned with the audio from the audiovisual material. Thus, by aligning the two audio signals, it follows that the text from the speech recogniser will be aligned with the video and audio from the audiovisual material.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in more detail, by way of example, with reference to the drawings, in which:

Figure 1 is a block diagram of an automatic subtitling system embodying the invention in its first aspect;

Figure 2 illustrates a method of allocating colours to speakers in accordance with an aspect of this invention;

Figure 3 illustrates a method of detecting scene changes in accordance with an aspect of this invention;

Figure 4 is a diagram showing an example of the use of the method of Figure 3;

Figure 5 is a block diagram of an automatic subtitling system in accordance with the preferred method embodying the invention;

Figure 6 is a block diagram showing more detail of a system for placing the subtitles in dependence upon the audio signal;

Figure 7 is a block diagram of a system which uses picture content, in particular information on faces, for placing the subtitles; and

alignment step, the recording of the programme, recorded on videotape for example, is replayed, step 18, and speech recognition forced or time alignment techniques are used to align the words from the script with the spoken dialogue from the replayed recording. This gives the required timings for each word of the text. The processed script from step 14 is also sent to a separate process, step 20, where colours are assigned for each of the individual speakers. As an alternative, shown in dashed line in Figure 1, the input to step 20 may be the script as output from the alignment step 16, that is after timing adjustment. Finally the subtitles are formatted in step 22, making use of the timing and text from the alignment step 16 and the colour assignment or allocation from step 20. This is assisted by the output of a shot change detector 24 which receives the replayed video signal from step 18 and generates an indication of whenever a shot change takes place. Shot changes take place relatively frequently and are relatively easy to detect compared with scene changes, which are discussed below. The output takes the form of a subtitle file 26 in accordance with the EBU standard.

For many broadcast programmes a script of the entire programme is normally available. The script will have been written before recording the programme and will have been updated during the production process to reflect any changes introduced.

Hence the task of creating subtitles for a programme is greatly simplified, as the text can be synchronised with the recorded dialogue to give each word its corresponding start and end times in the recording. Once these timings have been derived, the text is formed into

be sufficient words in common with the replayed audio signal.

The output signal from the microphone 40 is applied to a second input of the dynamic time warping block 34 and also to a speech recogniser 42. This may take the form of a commercially-available software speech recogniser such as Dragon's Naturally Speaking or IBM's Via-Voice (trade marks). The speech recogniser generates an electronic text signal indicated by block 44, which is stored in a text file.

The function represented by the dynamic time warping block 34 is to compare the timings of the two audio signals which are input to it, namely the audio signal from the audiovisual material and the microphone output signal. Dynamic time warping is of itself known and is described, for example, in "Digital processing of speech signals" by Rabiner and Schafer, ISBN 0-13-213603-1, see pages 480 to 484. The operation generates timings or timing differences for each word and these are stored in block 46. These timings are then used in a block 48 in the generation of the subtitles from the text from the speech recogniser. The timings are used to adjust the timing of the text in block 44 so that it is aligned with the audio from the original programme.

In the generation of the subtitles, other techniques described in our earlier application can also be employed.

The system illustrated in Figure 5 enables the rapid preparation of subtitles even in the absence of a typed text file, with minimum effort and time required.

interpreted as dialogue and attributed to the previous speaker.

These two difficulties mean that adopting an approach that uses pattern matching may not offer reliable results: parts of the script that are acting directions for example may be interpreted as names or dialogue if they contain typing errors. If errors such as this occur, then the subtitles produced will clearly contain incorrect text but will also be out of synchronisation with the dialogue around that region.

To overcome these problems we propose the use of a technique based on Bayesian statistics to analyse the format of the script and locate the elements needed. This approach will accept a variety of page layouts in the script and can adapt to new formats. It is also tolerant of typing errors. It is described more fully below.

Alignment (step 16)

The alignment stage (step 16) takes the speaker and dialogue information, or simplified script, that has been derived from the input script and, with reference to the audio from the programme, calculates timing details for each word spoken. This uses a known technique in speech recognition referred to as 'forced alignment', see for example Gold, B., and Morgan, N., "Speech and Audio signal processing", published by Wiley, 2000, ISBN 0471 35154-7, pages 348-349. Forced alignment, or, more generally, time alignment, is the process of using a speech recogniser to recognise each utterance in the dialogue but with reference to the expected text derived from the script. Hence the speech recogniser is working with a constrained vocabulary and grammar, which makes the task significantly

also take into account any colour assignments that the user has chosen to impose (such as the narrator may always be in yellow text) and work around these additional constraints.

Shot Change Detection (step 24)

The shot change detection stage in the process (step 24) takes the recording of the programme and applies shot change detection to the video. This produces a list of frames in the video where there is a change of shot, for example, switching to a different camera or viewpoint, or a change of scene.

Shot changes are helpful in subtitling because, if ignored, they can result in the subtitles being more confusing for the viewer to follow. If a subtitle is present when a shot change occurs in the video, there is a tendency for the viewer to look away from the text to the new image and then back to the text. By this point, they have lost their place in the text and re-read it from the beginning, which can leave insufficient time to read the entire subtitle before it ends. This can be avoided by ensuring that subtitles are either not present, or change, when the shot change itself occurs.

A further refinement to improve presentation, is that subtitles should not start or end near to a shot change, typically in a region 25 frames on either side. In this case, the subtitle in-time or out-time is changed to match the shot change. In this way there is less confusion for the viewer as the two change together.

A known shot change detection method can be used, for example one based on the mean average difference technique. See Ardebilian, A., et al., "Improvement of

achieve the best overall appearance of the subtitles. The basic steps in this process may for example be:

I) Add words, one-by-one, to form the subtitle, starting a new line as each one is filled.

ii) Colour the text as required for the speaker. If necessary, insert dashes or other markers into the text to identify different speakers using the same colour in the same subtitle.

iii) End the current subtitle optionally and start a new subtitle when there is a pause in the dialogue.

iv) When words are spoken as a shot-change occurs, end the current subtitle and optimise the grouping of words between the current subtitle and the next subtitle to minimise any interruption to the flow of reading. This must take into account the timing of the words, phrases and sentences, etc., the structure of the text, and when changes of speaker or scene occur.

v) Where possible, increase the duration of short subtitles (typically those less than one second in duration) to an acceptable minimum duration.

vi) Where possible, balance the text between the lines within each subtitle to achieve an even appearance on-screen.

can easily make the software produce subtitles of a quite different appearance. Hence we also achieve the flexibility that is attractive in a product.

The essence of the approach is to create subtitles from text using the following steps:

a) for each word in the script consider the options of adding it to the current line of the subtitle, starting a new line of text in the current subtitle or starting a new subtitle.

b) for each of these options, calculate a score for the subtitle being produced based on a number of attributes, e.g. number of lines in the subtitle, line length, position of the final word in its sentence, etc., (we use over 50 such attributes.)

c) continue to add each word in turn and re-calculate the subtitle scores

d) find the best sequence of subtitles that together maximise the overall score through the programme.

Step (c) is the where the algorithm is crucial in order to keep the search time to a minimum; here we are using a search technique similar to a well-known algorithm but modified for this application.

How many different sets of subtitles is it possible to make for a programme? If we consider that between each word and the next within a programme there can either be a continuation to the next word on the same line, a new line within the subtitle or a new subtitle, we reach the number: 3^n Continuations

For a 4000-word programme this is $3X10^{1908}$. Of course the actual number of possible subtitles is smaller as there is a limit on the number of words on a line and to the number of words in a subtitle, but there is

0, words 3, 4, 5, 6 and 7 in subtitle 1 and 8, 9 and 10 in subtitle 2).

There is a limit to the number of words that is possible to have in a subtitle. To simplify this explanation it has been set at 5. On the graph, this can be seen through the absence of a route along the x-axis past word 4. This is because such a route would represent having more than five words in subtitle 0, exceeding the limit. The limit also prevents a subtitle 1 from running beyond word 9, this is again visible on the graph, there are still some illegal routes possible on the graph that cannot be removed without also removing legal routes e.g., word 2 to word 10 in subtitle 2.

With all of this in mind, there are two things that we need to be able to do:

- Carry out the comparison of all these routes in an adequate time.
- Find out which of these routes (sets of subtitles) is best.

To do the first a system of penalties is introduced. Each route will incur a score (or penalty) so that the best scoring route can be chosen. The scoring system is pessimistic with a higher penalty (score) indicating a less optimal subtitle. A fixed penalty can be introduced at each location in the graph for continuing and another penalty at each point for splitting to a new subtitle. We then look for the route with the lowest overall penalty.

To produce well formatted subtitles we may, for example, have a high penalty for splitting anywhere but at the end of a sentence and a high penalty for continuing from a word in one scene to one in the next. The advantage of having penalties associated with splitting

[Best route to Word 0] (Word1)

Word 0 is in square brackets because we don't care what comes before word 1, we simply chose the best route up to the end of word 0. (In this case there is no difference as there's only one possible route)

The best of the above two subtitles is picked and stored.

At word 2:

First we try all possible routes from word 0 without splitting:

(Word0 Word1 Word2)

Now we try all possible routes which split between words 0 and 1

[Best route to Word 0] (Word1 Word2)

And then all possible routes which split between words 1 and 2

[Best route to Word 1] (Word2)

From these we can store the best as the best possible route to word 2. This process continues on through the programme. Assuming we have a maximum subtitle length of 5 words, the situation at word 10 will be as follows:

The subtitle (Word0 Word1 ..... Word10) is not considered because it is illegal, as is the route represented by [Best route to Word 0] (Word1 ..... Word10). The starting point has to be at word 6 (as '6,

[Best route to Word 6] (Word7 Word8 Word 9)

One new Line:

[Best route to Word 6] (Word7 [new line] Word8 Word 9)

[Best route to Word 6] (Word7 Word8 [new line] Word 9)

Two new lines

[Best route to Word 6] (Word7 [new line] Word8 [new line] Word 9)

While this results in considerably more combinations than the one line model, it still shrinks the search space for a 4000 word programme from 3 $x10^{1908}$ to $2.6X10^{6}$ comparisons.

This subtitling algorithm reduces the history from a given word in the programme, back to the start, to a single routee. This can be shown in a diagram similar to a state machine in figure 11. The difference here is that all of the routees to a subtitle starting at word n are shown to converge (because only one of them will be stored). A significant difference between this diagram and the state machine of figure 10 is that the gradient of the bottom of the graph is the same as that at the top. This shows that the search space expands in a linear fashion as words are added.

At each word there is a penalty for continuing to the next word, a penalty for splitting to a new subtitle. These penalties are worked out before the search is performed and are outlined in the pre-processed scoring section below. Some additional scoring takes place at run

other prefer to split to a new subtitle. To accommodate the former the penalty for splitting to a new line at the end of a sentence should be low, whereas for the latter it should be high and the penalty for splitting to a new subtitle low.

The less fragmented text is, the easier it is to read so it's best to continue to the next word in a sentence on the same line if possible. If a sentence won't fit on a single line then it is easier to read it divided over a few lines (in one subtitle) than it is to read the same sentence divided into a few subtitles. It follows then, that to make subtitles easy to read, the penalties for splitting to a new subtitle should be higher than for splitting onto a new line.

All penalties are relative so the actual numbers have no significance. As a rule of thumb, new subtitle penalties should be 2 to 5 times the new line penalty for the same location. The continuation penalty can be set for the end of a sentence or speaker. This is usually left at 0 unless the subtitler wants to split to a new subtitle at the end of each sentence/speaker.

Penalties for splitting within a speaker follow the same model as splitting within sentences. If the minimum penalty is set to a relatively high level but the gradients are less steep than that for sentences then the lowest penalty should be obtained by trying to keep as much of a speaker's dialogue as possible in a single subtitle. If this is not possible, then splitting at the end of a sentence should get the lowest penalty. Splitting in the middle of a sentence is only likely when a sentence is too long to fit into one subtitle.

top line being shorter than the bottom line and vice versa are independently defined so that a preference for subtitles like this:

(Formatting subtitles
isn't as easy as it looks)

Can be expressed over subtitles like this:

(Formatting subtitles isn't
as easy as it looks)

Studies suggest that the first of these examples is easier to read, as your eye doesn't have to move as far. It also uses up most space at the bottom of the television picture, where it is less likely to obscure something important.

For three-line subtitles a mean of the line length is calculated and then the penalty is proportional to the sum of differences between each line's length and the mean. A penalty is also added for subtitles that are concave:

(Formatting subtitles
isn't
as easy as it looks)

This only applies to subtitles where the middle line is at least 8 characters shorter than the lines above and below it. At present there is no scoring based on the shape of one-line subtitles.

A simple penalty for each empty letter on a line in a subtitle exists. Setting this promotes wide subtitles.

this is then used to affect the justification of the text for each speaker so that their text matches their position in the picture.

Generation of subtitles - placement using stereo audio signals

The system 60 shown in Figure 6 starts with the programme soundtrack 62. This is applied to a band-pass filter 64 which typically passes the frequencies 700Hz to 5kHz. These frequencies are passed to a circuit 66, which determines the relative powers of the left and right components of the stereo signal. The detailed construction of such a circuit is well within the competence of those skilled in the art and the precise construction will generally depend on the equipment in which it is to be used. The relative power measurement can be relatively coarse as all that is required in a simple embodiment is to determine whether the left and right signals are roughly equal, and if not, which is the greater of the two.

This relative power measurement is then applied to a position analysis circuit 68 which converts the relative power measurement into a subtitle position signal. Typically, this has three values, namely "left", "centre", and "right". This position signal is used by the subtitle-formatter 22 (see Figure 1 of our earlier application).

If a more sophisticated method of measuring relative power is used, the subtitle position signal can be arranged to represent intermediate positions and not just left, centre and right.

While the use of the tracking of faces and/or lip movement has been described, other items may be tracked such as a human silhouette or items of clothing or jewellery, for example.

Subtitle re-versioning

Another application for the system described is termed 're-versioning' of subtitles. In this case subtitles will have already been prepared by the programme producer or original broadcaster. However, when the programme is broadcast on a different channel, typically by another broadcaster and in a different country, the original subtitles may not be entirely appropriate. As well as not matching the preferred style of the second broadcaster, they may no longer be correctly synchronised to the programme because of timing differences introduced by the conversions that occur from film and video tape between the differing television standards. When this occurs, each subtitle must be re-synchronised to its dialogue, a process that is again very time-consuming.

A system embodying the present invention can solve this problem very effectively but, in this case, the original subtitles are used instead of the script to provide the text of the speech. The original subtitles may have been prepared by the above-described method. Once the alignment phase has been completed, the user has the option of either using the new timings to produce replacement in-times and out-times for the original subtitles, or generating completely new subtitles from the text of the original subtitles (i.e. as though they had been the script). In this latter case, speaker details would not be available but speaker identification applied

The analysis undertaken by the circuit 88 can be improved in reliability by making use of punctuation from the subtitle text, so as to cluster together words into longer sequences or phrases that are most likely to be spoken by the same speaker. Reference is made to "Digital processing of speech signals", by Rabiner and Schafer, ISBN 0-13-213603-1, pages 485 to 489 in relation to speaker identification techniques and appropriate speech parameters.

When a change of speaker is identified, the circuit 88 provides the new speaker's voice parameters to a circuit 90. The circuit 90 stores the voice parameters for each of the speakers so far identified in the programme, and compares the parameters for the new speaker with the stored parameters. In this way the circuit 90 determines whether the new speaker's parameters are substantially the same as the stored parameters for one of the previous speakers, and if so assumes that the 'new' speaker is, in fact, that previous speaker again. If a significantly different voice is detected, then this is interpreted as a new speaker. The new speaker's parameters are now themselves stored, to update the details of the speakers, in a stage 92. Thus, the system counts the number of speakers in the programme.

Occasional speaker identification errors may occur but in practice the system operator will generally work through the subtitles as the analysis proceeds and so can correct any speaker identification errors as they occur. If this is done the stored parameters for the previous speakers are reliable, and hence identification of subsequent speakers becomes more reliable as the operation continues.

with a link grammar", Third International Workshop on Parsing Technologies, August 1993.

Live Subtitling

In a live-subtitling situation, such as a news programme, there can be sections where the text is scripted beforehand, e.g. the news presenter's introduction to each story. For these parts, the text can be turned into subtitles before it is spoken as these can be held ready for transmission and then output as the presenter reaches the corresponding point in the scripted text. Currently this is done by hand.

Some elements of the present invention could also be used to automate this process. The text can be formed into subtitles automatically, although shot changes and timing details would not be available at that point. The forced alignment technique can be used to track through the script as the words are spoken in the broadcast and as the first word of each pre-prepared subtitle is spoken, that subtitle can start to be transmitted.

Detail of Script Processing using Bayes' theorem

We propose a script-processing procedure that can decode script files, using a statistical description of the script format. These descriptions are contained in *probabilities files*. They can be written for specific programmes or for more general script formats.

The system thus analyses scripts in a statistical way. The script is first broken down into *blocks*, to separate text in different styles and at different positions on the page. Then, for each block, probabilities are evaluated for a number of hypotheses as to the type of

The chosen hypothesis for the previous block when on the previous line.

The chosen hypothesis for the previous block.

The chosen hypothesis for a block directly above the current block.

The block being in the most likely column for a particular type of text.

Finally, there are properties that test attributes in other blocks and ones that combine properties:

Any property, tested on the next text block.

Any property, tested on the previous text block.

Any property, tested on the whole line containing the current block.

Any property, tested on the first character of the current block.

Any property, tested on the first word of the current block.

Both of two properties being true.

Either of two properties being true.

To differentiate between independent and non-independent properties, the various properties listed in the *.prob* file are grouped. Each group is considered to be independent of the others but will contain non-independent properties. Foe each property within a given group, the *.prob* file specifies the likelihood that the property is true and that those before it in the group are false. Thus the likelihoods relate to mutually-exclusive events within the group.

As noted previously, the approach based on Bayesian statistics can adapt to new formats. Adaptation enables the script processing software to cope with a variety of script formats without having a statistical description of each one individually. The software processes the script initially according to the likelihood values listed in the *.prob* file. Then, for properties marked for adaptation in the *.prob* file, new statistics are estimated from the results of the initial analysis, using Bayes' theorem once again. This step reduces the mis-classification of blocks of text in the script.

Using this technique, a generic statistical description covering a variety of script formats can be prepared. For example, on a first pass, the analysis might mis-classify a small proportion of speaker names, perhaps due to typing errors or inconsistencies in the script. However, if the majority of the text has been classified correctly, the re-estimation process will pick up other properties that distinguish the different text types, leading to a more accurate classification on the second pass.

Detail on assigning colours to characters

Typically, a number of rules are applied when choosing colours for subtitles, such as:

Each speaker must keep the same colour throughout the programme.

Colours other than white can only be assigned once per scene.

The last speaker of one scene should not use the same non-white colour as the first speaker in the next.

algorithm therefore includes an empty 'null' grouping in the list of possibilities (there may be more than one).

In Figure 2, in stage 1 shown at the left of the Figure, from an initial list the last group is selected or marked. The term 'group' here covers just a single speaker. A search is then conducted downwards from the marked group for possible combinations. In stage 2, the previous marked group, in this case 'C', is considered. The search now finds that 'D' can be combined. A new group 'CD' is created. In stage 3 the procedure is repeated and three more groups are found, namely 'BC', 'BD', and 'BCD'. Stage four shows that further groups containing 'A' are also found. The double-headed arrow on the figure shows the region to search at each stage.

Each of the available colours is then allocated to one of the groups. In most cases, no set of four groupings will completely cover all the speakers. In this case, the leftover ones must share 'white', and any interactions involving these speakers and the other 'white' speakers will require leading dashes to highlight the change of speaker.

The colours are assigned to the groups of characters as follows.

With four colours, there could be in excess of 10^{12} potential colouring schemes so it is clearly not practicable to search them all. Fortunately, it is possible to identify groupings that are likely to contribute to a good colouring scheme and thus reduce the search space dramatically.

A grouping whose speakers have a large number of interactions with other speakers is a 'good' candidate for having a colour allocated to it. It is also necessary to

the number of words coloured white (the more the better, for readability).

By sorting the groupings list with consideration of the above points, very good results can be obtained from a search of only a few thousand schemes. Although this cannot guarantee to find the best colouring scheme, experiments have shown that extending the search (at the expense of longer processing times) only gave very small improvements.

Because this algorithm searches a number of possibilities, it can be adapted to use any definition of a 'good' colouring scheme.

Improvements on the basic colouring scheme

One aim of the colouring scheme is to minimise the number of subtitles in which more than one speaker's words are coloured white. As described, the colouring algorithm considers all interactions between speakers when considering candidate colour schemes. An improvement can be made by only considering those interactions which are likely to end up in one subtitle. By performing the colouring process after the alignment process has been completed, interactions that are separated in time can be ignored and a more optimal colouring scheme can be found.

The groupings algorithm described above also has some limitations. Firstly, it does not cater for manual colour assignments and, secondly, it can occasionally find itself swamped by millions of candidate groupings. 'Manual assignments' means that the user should be able to override the colours for one or more of the speakers, that is, specify a predetermined colour for them. The algorithm can be improved to allow this as follows.

AB, C, D, E

AB, C, DE, (null)

AB, D, CE, (null)

AB, CDE, (null), (null)

ABCE, D, (null), (null)

Out of these, the software would probably select the third option, with white as the third colour since white text is more readable.

Some programmes have a large number of speakers with very little interaction between them. This situation can result in a huge number of possible groupings, consuming vast amounts of memory. For example, there may be 40 or so speakers in total but rarely more than three together at any one time. There is little interaction outside these small groups. The general problem is where there are many speakers with few 'clashes'. To overcome this problem, any speaker with fewer clashes than there are colours need only be given a colour once the other speakers have been satisfied. Thus, with little extra processing, such speakers can be left out of the groupings list and assigned colours in a new final stage.

Scene change detection

A starting point for finding 'scene changes' is a graph of speaker against time. Such a graph might appear as shown in Figure 3. In this figure, the dotted lines indicate logical places for scene changes where the group of active speakers changes. Determining these first of all requires filtering horizontally to ignore individual interactions.

We propose the use of an algorithm which effectively constructs a matrix representing the diagram of Figure 3

to insert a scene change so as to produce ABACAD in one scene and ADEDA in the next; speaker D then appears in both scenes. The best place to put the scene change is where the number of distinct speakers to the left plus the number on the right is at a minimum. For the example given, this value is 4+3=7. Inserting the scene change in the 'correct' place to give ABACA and DADEDA gives a value of 3+3=6. The software therefore preferably searches a certain amount both ways from an initial estimated scene change, looking for a minimum in this value.

As an example, Figure 4 represents the first 900 words of a television programme. The horizontal axis shows the word number and the vertical positions of the black rectangles indicate the different speakers, as before. The superimposed curve is the value of the 'scene change indicator' function and the vertical lines show the positions of the detected scene changes. This particular programme does indeed have scene changes at every point shown. However, the algorithm has missed the first scene change, which occurs at about 80 words. This is due to it being very close to the beginning of the programme when compared to the Gaussian window width of about 300 words. However when the algorithm is used to assist colour allocation, the actual accuracy of the scene change detection is not of paramount importance; all that is required for good colouring is a set of scene changes that is reasonable. However, having too few scene changes will cause more white to white interactions due to reduced colour availability. Too many will allow colours to be re-used quickly and this may confuse the viewer.

CLAIMS

1. A method of generating subtitles for audiovisual material, comprising the steps of:

receiving and analysing a text file containing dialogue spoken in the audiovisual material to provide text information signal representative of the text;

aligning the text information and the audio signal from the audiovisual material in time using time alignment speech recognition to provide timing information for the spoken text; and

forming the text information and the timing information into an output subtitle file.

2. A method according to claim 1, in which the step of analysing the text file comprises calculating with the use of Bayes' theorem probabilities that each of a plurality of blocks of text is one of a plurality of text component types.

3. A method according to claim 1, in which the step of analysing the text provides a text information signal representative of the text and of the person speaking the text.

4. A method of assigning colour representative of different speakers to subtitles, the method comprising the steps of:

forming a plurality of groups of speakers, where each group contains speakers who can be represented by the same colour; and

assigning the available colours to a corresponding number of the plurality of groups, the groups being selected such that all the speakers are allocated a colour.

10. A method according to claim 9, in which the detecting step includes the step of filtering in time with an averaging function.

11. A method of parsing an electronic text file to identify different components thereof, comprising the steps of:

identifying blocks of text in an input electronic text file;

providing a plurality of possible script format properties for the blocks;

providing a definition of each of the possible components of the text file;

in relation to each block, determining the value of each script format property;

for each block, determining from the script format properties of the block and the component definitions a probability value that that block is each of the component types;

selecting the component type for each block on the basis of the probabilities that it is each of the component types; and

generating therefrom an output file.

12. A method according to claim 11, in which the step of determining probability values is undertaken using Bayes' theorem.

13. A method according to claim 11, in which the output file is input as a new input file and the processing repeated.

14. A method according to claim 11, in which the component definitions are adaptively changeable.

20. A method according to claim 1, in which the text file is generated by:

playing the audio signal from the audiovisual material, the audio signal containing speech;

having a person listen to the speech and speak it into a microphone; and

applying the microphone output signal to a speech recogniser to provide an electronic text file.

21. A method of placing subtitles related to speech from speakers in a moving picture, comprising the steps of:

receiving a video signal representative of the picture;

analysing the video signal to identify areas of the picture which indicate the presence of a speaker in a location on the picture;

generating therefrom a signal which indicates a desired location for a subtitle relating to speech spoken by that speaker; and

placing the subtitle for that speaker in accordance therewith.

22. A method according to claim 21, in which the analysing step comprises identifying faces and/or lip movements.

23. A method of generating subtitles for audiovisual material, comprising the steps of:

receiving a text signal containing text corresponding to speech in the audiovisual material;

identifying from the audio signal from the audiovisual material predetermined characteristics of the speakers voice;

deriving a set of subtitles from the text information signal; characterised in that the deriving step comprises:

a) considering each word in turn in the text information signal;

b) assigning a score to each subtitle in a plurality of different possible subtitle formatting options leading to that word;

c) repeating steps a) and b) until all the words in the text information signal have been used; and

d) deriving the subtitle formatting option that gives the best overall score for the text information signal.

30. A method according to claim 29, including the step of storing the subtitle formatting option giving the best overall score to at least one selected point in the text and performing step b) only on words added from that at least one selected point.

31. A method according to claim 30 in which the position of the at least one selected point changes position as words are added, thereby reducing the number of subtitle formatting options for which scores must be derived.

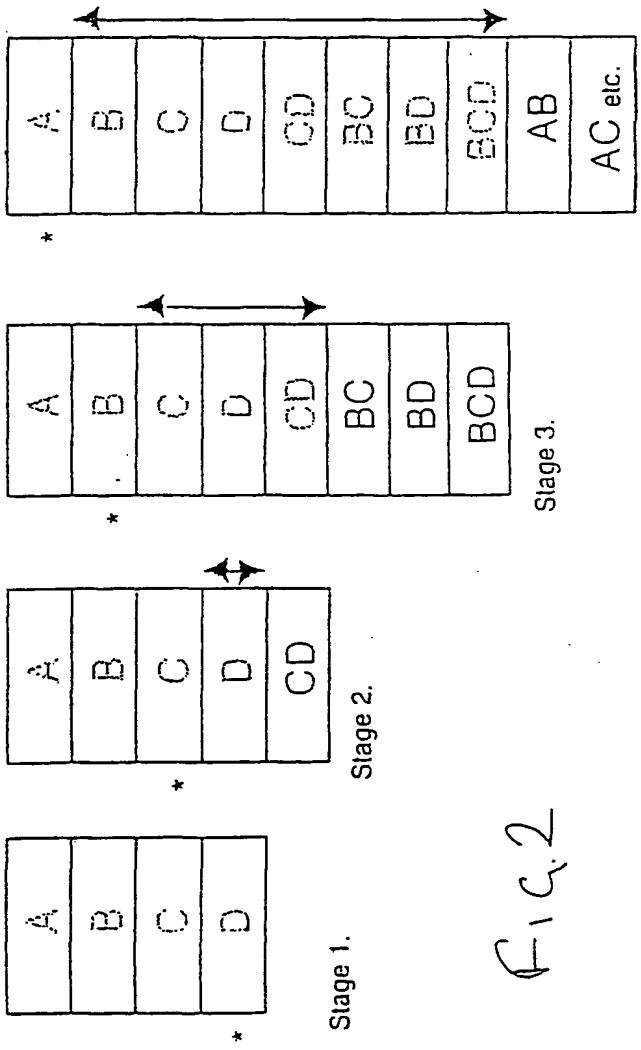

FIG 2

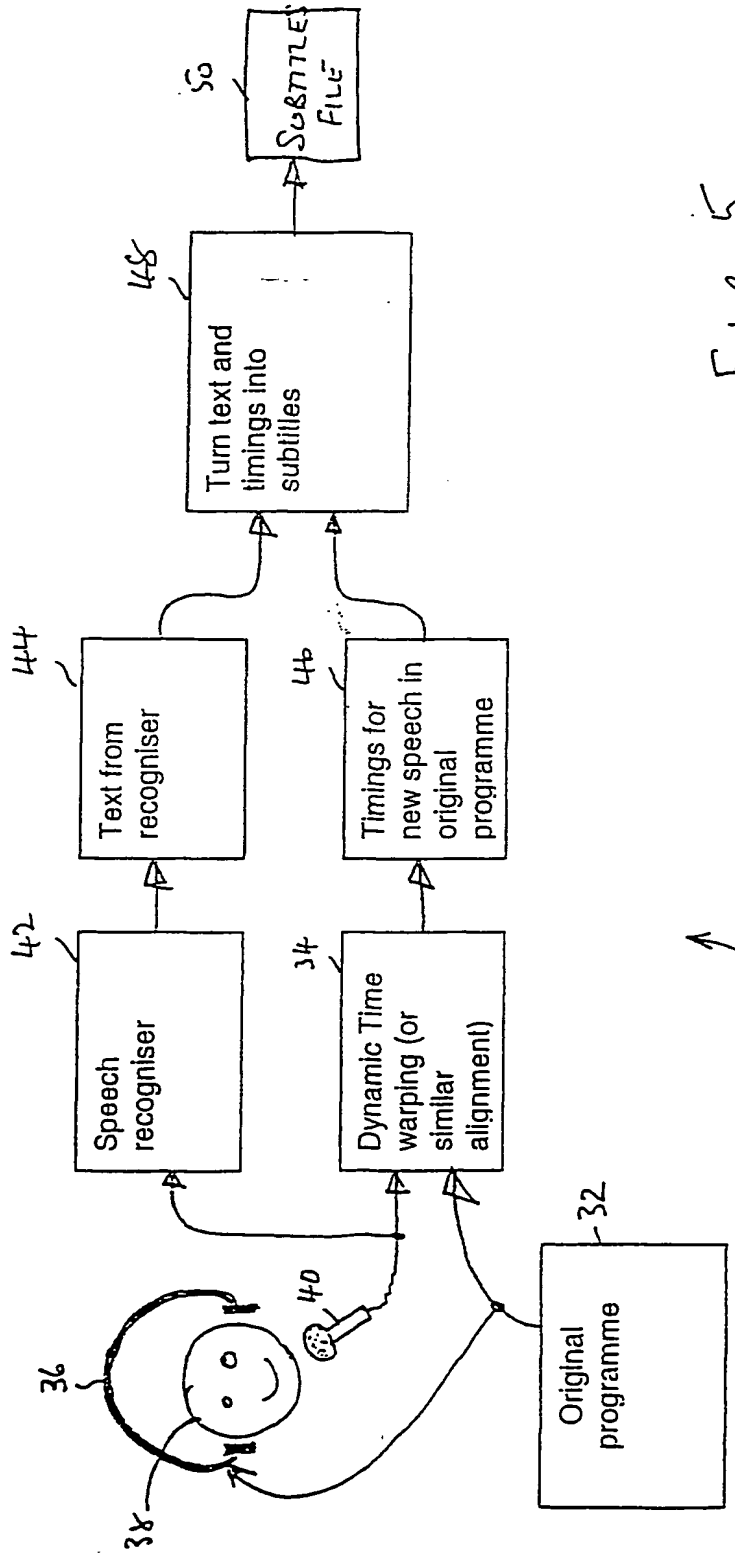
Original programme
32
Speech recogniser
42
Dynamic Time warping (or similar alignment)
34
Text from recogniser
44
Timings for new speech in original programme
46
Turn text and timings into subtitles
48
Subtitle file
50
30
36
38
40

FIG. 5

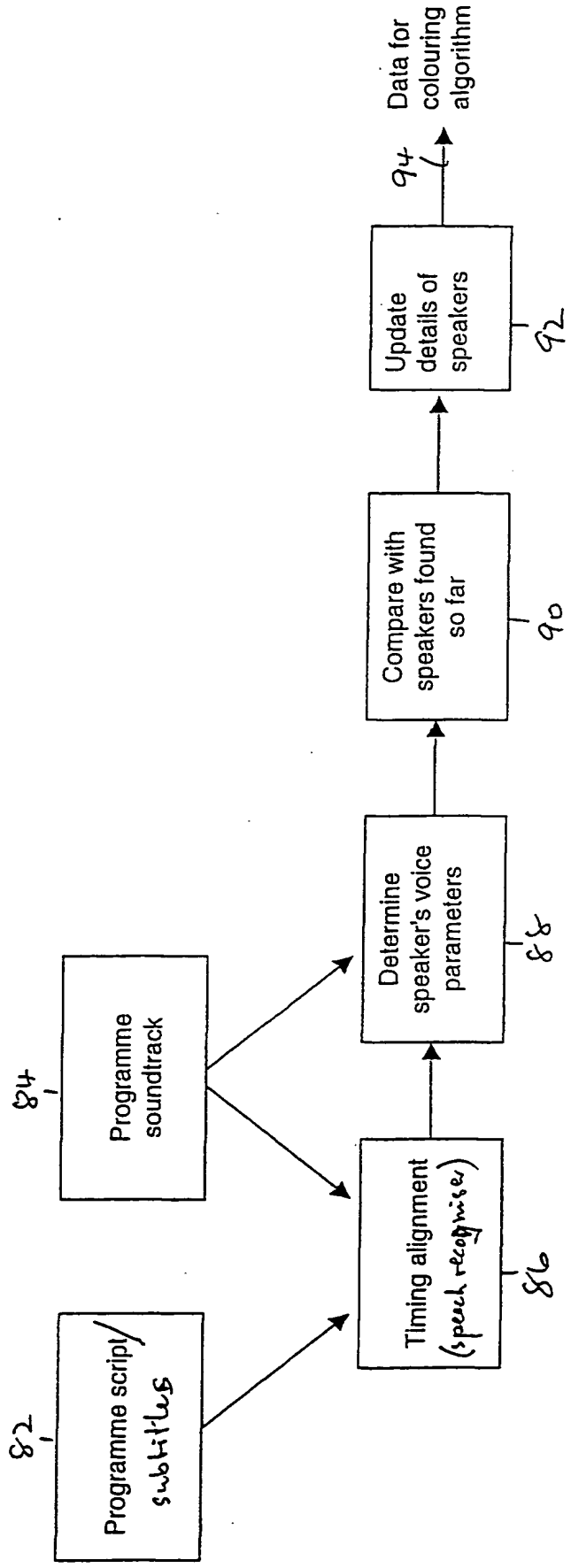
Programme script/ subtitles
82
Programme soundtrack
84
Timing alignment (speech recogniser)
86
Determine speaker's voice parameters
88
Compare with speakers found so far
90
Update details of speakers
92
Data for colouring algorithm
94
80

FIG. 8

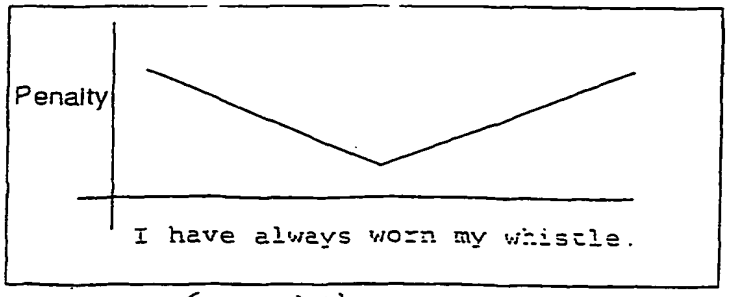

Figure 12

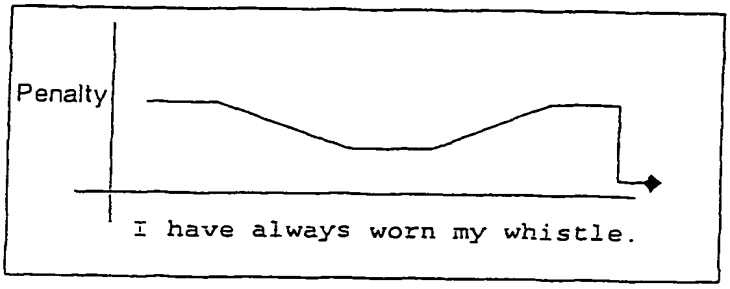

Figure 13

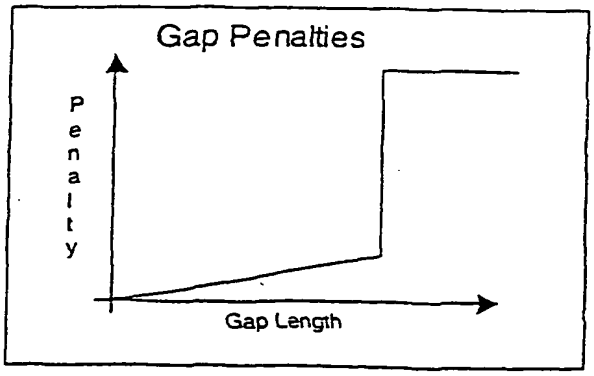

Figure 14